Risk-Conditioned Reinforcement Learning: A Generalized Approach for Adapting to Varying Risk Measures

Gwangpyo Yoo[1], Jinwoo Park[2], Honguk Woo[1]*

[1]Department of Computer Science and Engineering, Sungkyunkwan University
[2]Department of Artificial Intelligence, Sungkyunkwan University
necrocathy@skku.edu, pjw971022@skku.edu, hwoo@skku.edu

Abstract

In application domains requiring mission-critical decision-making, such as finance and robotics, the optimal policy derived by reinforcement learning (RL) often hinges on a preference for risk management. Yet, the dynamic nature of risk measures poses considerable challenges to achieving generalization and adaptation of risk-sensitive policies in the context of RL. In this paper, we propose a risk-conditioned RL framework that enables rapid policy adaptation to varying risk measures via a unified risk representation, Weighted Value-at-Risk (*WV@R*). To sample risk measures that avoid undue optimism, we construct a risk proposal network employing a conditional adversarial auto-encoder and a normalizing flow. This network establishes coherent representations for risk measures, ensuring the monotonicity of the quantile representations of risk measures. Through experiments with locomotion, finance, and self-driving scenarios, we show that our framework is capable of adapting to a range of risk measures, achieving comparable performance to the baselines individually trained for each measure. The framework often outperforms the baselines, especially in the cases when exploration is required during training but risk-aversion is favored during evaluation.

1 Introduction

Reinforcement learning (RL) offers a promising approach to various sequential decision-making problems found in real-world applications, including robotics, finance, and autonomous driving. This is achieved by maximizing the rewards obtained from these environments. Many of these tasks inherently possess uncertainty, and the behavior of RL agents varies depending on the evaluation criteria used to measure this uncertainty. We refer to this (aleatoric) uncertainty as *risk* and to the evaluation criteria that consider this uncertainty as *risk measures*. We also refer to the risk measure's degree of optimism as *risk preference*. For example, conditional value at risk (*CV@R*) with confidence $\alpha \in [0, 1]$ is a risk measure calculated by taking the average from the most pessimistic return to the α-th quantile.

The choice of an appropriate risk measure is crucial in that it influences the agent's behavior, which can lead to undesired outcomes such as physical harm. However, determining the appropriate risk measure is challenging because a universal solution may not exist. Moreover, optimal risk measures can vary significantly across different applications or among individual users. For instance, in the area of autonomous driving, the selection of risk measures should take into account various factors such as time constraints, fuel levels, and the specific needs of passengers. A self-driving agent for delivery service might prioritize a risk measure that emphasizes time constraints, whereas a family-oriented service, especially one transporting young children, might place a premium on safety above all else. In the realm of finance, risk measures should be chosen according to interest rates, prevailing market conditions, and a client's unique risk preferences (Acerbi, Nordio, and Sirtori 2001; Acerbi and Simonetti 2002; Adam, Houkari, and Laurent 2008; He, Jin, and Zhou 2015; BCBS 2009). For example, when the base rate increases, individuals tend to be risk-averse and vice versa.

For an RL agent to adapt to varying risk measures, it should be conditioned based on varying risk measures rather than being optimized for a single specific risk measure. Choi et al. (2021) proposed RCDSAC, a risk-conditioned RL framework, which adapts to specific ranges of risk measures, defined by simple parameters. However, the ranges of risk measures that RCDSAC can accommodate are relatively specific, thereby limiting its applicability. For example, *CV@R* and *median* are considered essential risk measures in both finance and non-finance areas (Zhang et al. 2018; Pruzzo, Cantet, and Fioretti 2003; Jabr 2005; Rahimi and Ghezavati 2018; BCBS 2009), but RCDSAC is not designed to handle these risk measures simultaneously.

Our goal is to expand the range of risk measures for risk-conditioned RL agents, based on the weighted value at risks (*WV@R*). *WV@R* represents a comprehensive set of risk measures with varying degrees of importance from the most pessimistic to the most optimistic returns. It is defined as the weighted integral over the quantile function of returns (He, Jin, and Zhou 2015). This universality allows risk-conditioned agents to simultaneously handle diverse risk measures.

To train a risk-conditioned agent using *WV@R* within the context of RL, it is crucial to appropriately represent risk measures as inputs to the agent's neural network. To access the RL loss with respect to these risk measures, it is also required to conditionally sample risk measures from *WV@R*

*Honguk Woo is the corresponding author.

in the quantile form. As excessively optimistic risk measures might have a detrimental impact on training (Kuznetsov et al. 2020), it is important to control over risk preference settings. To address these challenges, we introduce a risk proposal network, designated to conditionally generate risk measures from the *WV@R* set in accordance with risk preferences.

While generating risk measures, we often encounter the crossing quantile problem where the estimated quantile function violates its fundamental monotone property. A risk-conditioned agent struggles in learning from *WV@R* risk measures due to the semantic (importance) distortion by the crossing quantile. See Figures 2 and 7 for the problem.

To tackle the problem, we employ a non-crossing quantile regression approach that uses normalizing flows. This approach employs a conditional adversarial auto-encoder (AAE) as a generative model to produce proper *WV@R* risk measures. Consequently, the agent's actions can be aligned according to these risk measures.

Throughout several experiments with self-driving and finance case studies, our approach demonstrates that risk-conditioned RL agents trained through the risk proposal network achieve robust performance comparable to IQN (Dabney et al. 2018), a state-of-the-art risk-sensitive RL framework. This achievement is remarkable, considering that our approach is not tailored to specific evaluation risk measures, but universal in nature. We posit that this performance advantage stems from more effective exploration strategies, as elaborated in (Zhou, Wang, and Feng 2020).

Our contributions are as follows:

- We introduce a novel framework for risk-conditioned RL agents using *WV@R* that encompasses a wide range of diverse risk measures.
- We devise the risk proposal network to effectively embed risk measures and generate them.
- We also provide the non-crossing quantile regressor to ensure the semantics of generated risk measures.
- We demonstrate the generality and superiority of our approach with several case studies.

2 Related Work

Risk-sensitive RL: In the field of risk-sensitive RL, traditional research has focused on optimizing RL agents for specific risk measures (Howard and Matheson 1972; Sato, Kimura, and Kobayashi 2001; Mihatsch and Neuneier 2002; Tamar, Glassner, and Mannor 2015; Chow et al. 2017; Dabney et al. 2018; Vadori et al. 2020). Early efforts include optimizing an instance of *WV@R* like *worst-case* (Mihatsch and Neuneier 2002) or *CV@R* (Tamar, Glassner, and Mannor 2015; Chow et al. 2017; Dabney et al. 2018).

Notably, the IQN framework (Dabney et al. 2018), which bridges the distributional RL and risk-sensitive RL, stands out for its ability to compute the *WV@R*-based risk measures by estimating a quantile function of a policy's returns. RCDSAC (Choi et al. 2021) extends the application of risk-sensitive RL to risk-conditioned RL within the IQN framework. It focuses on a risk measure that could be parameterized as subsets of *WV@R* (e.g., *CV@R*, *CPW*), and achieves the risk-conditioned objective through a uniform sampling of these parameters. However, in this approach, risk measures whose parameter spaces are not shared cannot be addressed simultaneously. In our work, this limitation is addressed by introducing an encoder for risk measures in *WV@R*.

Crossing Quantile Problem: The crossing quantile problem is a significant concern in both statistics (Bondell, Reich, and Wang 2010; Dette and Volgushev 2008) and machine learning areas (Brando et al. 2022; Zhou, Wang, and Feng 2020). In the field of statistics, Brando et al. (2022) tackled the problem by exploring the numerical integration over a non-negative neural network, similar to UCMNN (Wehenkel and Louppe 2019). Yet, it is notably limited by its high computational cost, due to the exhaustive numerical integration.

In the RL literature, Zhou, Wang, and Feng (2020) introduced the non-crossing QR-DQN as a solution to the crossing quantile problem, highlighting the significance of exploration strategies. However, it faces compatibility issues with IQN. This incompatibility lies in its reliance on static points of the domain (i.e., quantile-level or cumulative probabilities) for quantile estimation, in contrast to IQN that samples such points randomly. As a result, the non-crossing QR-DQN is not well-suited for applications in risk-conditioned RL, where diverse sampling strategies for quantile-levels are demanded.

3 Problem Formulation

3.1 Distributional RL

We introduce our problem's domain, a Markov Decision Process (MDP), represented as $(\mathcal{S}, \mathcal{A}, r, \mathcal{P}, \gamma)$. Here, $\mathcal{S}$ denotes the state set, $\mathcal{A}$ the action set, $r(s_t, a_t)$ the reward for given $(s_t, a_t) \in \mathcal{S} \times \mathcal{A}$, $\mathcal{P}(s_t, a_t, s_{t+1}) = \Pr[s_{t+1}|s_t, a_t]$ the transition probability, and $\gamma \in [0, 1)$ the discount factor (Puterman 2014). For an agent following policy π, the discounted future return is represented as Q^π; $Q^\pi(s, a) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$, where $s_0 = s$ and $a_0 = a$, respectively.

To estimate the value under the *WV@R* risk measure, we need to obtain the quantile of Q^π. In particular, we focus on distributional RL that infers the quantile of Q^π. Distributional RL aims to estimate the distribution of Q^π rather than its expectation $\mathbb{E}[Q^\pi]$ (Bellemare, Dabney, and Munos 2017). Distributional RL algorithms achieve the goal by constructing a distributional version of temporal difference (t.d.) loss. We introduce the following definition to formalize the concept.

Notation 1. *From now on, we will use the symbol x to represent quantile-levels (domain points) for quantile functions; e.g., x_i, $\hat{x}_i$, x will belongs to $[0, 1]$ and all of them represent quantile-levels.*

Definition 1. *A quantile of random variable Q is defined as the inverse function $F_Q^{-1} : [0, 1] \to \mathbb{R}$ of the cumulative density function of random variable $F_Q : \mathbb{R} \to [0, 1]$. We write $\overleftarrow{Q}(x)$ instead of $F_Q^{-1}(x)$, whenever clear. For a conditional random variable Q with respect to variables s, a, we write $\overleftarrow{Q}(x; s, a)$ instead of $F^{-1}_{Q|s,a}(x)$.*

To estimate the quantile function from data, we also introduce the quantile regression loss.

Definition 2. *Let $\{Y_j\}_{j=1}^{K'}$ be $K' \in \mathbb{N}$ numbers of realized random variables to estimate (i.e., ground truth) and*

$\{\overleftarrow{Y}_\theta(x_i)\}_{i=1}^K$ be the estimated x_i-th quantile where $K \in \mathbb{N}$. The quantile regression loss is an asymmetric L^1 loss such as

$$\mathcal{L}_{QR}(\delta_{ij}) = \underbrace{\frac{1}{K'K}\sum_{j=1}^{K'}\sum_{i=1}^{K}}_{\text{average for each pair}} \underbrace{|x_i - \mathbb{I}(\delta_{ij} < 0)|}_{\substack{\text{asymmetric weight} \\ \text{based on sign}(\delta_{ij})}} \cdot \underbrace{|\delta_{ij}|}_{\substack{\text{pair-wise} \\ \text{difference}}} \quad (1)$$

where $\delta_{ij} = Y_j - Y_\theta(x_i)$ and $i = 1, \ldots, K, j = 1 \ldots K'$.

We also introduce the distributional Bellman-optimality equation (Bellemare, Dabney, and Munos 2017) defined as

$$Q^\pi(s_t, a_t) \overset{\text{distr.}}{=} r(s_t, a_t) + \gamma Q^\pi(s_{t+1}, a) \quad (2)$$

where $a = \text{argmax}\mathbb{E}_{a_{t+1}\in\mathcal{A}}[Q^\pi(s_{t+1}, a_{t+1})]$. To obtain the optimal Q^π's quantile estimator, denoted by $\overleftarrow{Q}_\theta$, we minimize the quantile t.d. loss analogous to conventional RL's t.d. loss. The quantile t.d. error is defined as

$$\delta_{ij} = \underbrace{r(s_t, a_t) + \gamma \overleftarrow{Q}_\theta(x_j; s_{t+1}, a)}_{\text{(Bellman-TD) target } Y_j \text{ in Def.2}} - \underbrace{\overleftarrow{Q}_\theta(x_i; s_t, a_t)}_{\overleftarrow{Y}_\theta(x_i) \text{ in Def. 2}} \quad (3)$$

where a is that of Eq. (2). Applying δ_{ij} to Eq. (1), we obtain the quantile t.d. loss. Note that sampling strategies for x_i, x_j are algorithm dependent; e.g., TQC (Kuznetsov et al. 2020) and IQN have different sampling strategies.

To define a risk-sensitive objective, we formalize *WV@R* as below.

$$WV@R(Q; \phi) := \int_0^1 \overleftarrow{Q}(x)\phi(x)dx = \mathop{\mathbb{E}}_{x\sim\phi}\left[\overleftarrow{Q}(x)\right] \quad (4)$$

The rightmost term is from the inverse sampling transform and $\phi : [0, 1] \to [0, \infty]$ is a probability density function on $[0, 1]$ which defines *WV@R* risk measures. Since ϕ governs Eq. (4), we refer to ϕ (or its quantile $\overleftarrow{\phi}$) as a risk measure. The risk-sensitive RL objective for a fixed ϕ is to find the optimal policy π^* such that

$$\pi^* = \text{argmax}_\pi \mathbb{E}[\overleftarrow{Q}(x; s, \pi(s))]. \quad (5)$$

If we replace a with $\text{argmax}\mathbb{E}_{x_j\sim\phi}[\overleftarrow{Q}(x_j; s_{t+1}, a)]$ in Eq (2), we establish a risk-sensitive RL agent satisfying Eq. (5) (Dabney et al. 2018), which conducts risk-sensitive actions instead of risk-neutral ones.

3.2 Our Objective

We formalize our objective as follows. Let $\pi : \mathcal{S} \times \Phi \to \mathcal{A}$ where $\Phi = \{\phi : [0, 1] \to [0, \infty] | \int_0^1 \phi(x)dx = 1\}$ is the set of risk measures. Let $\overleftarrow{Q^\pi} : [0, 1] \times \mathcal{S} \times \Phi \times \mathcal{A} \to \mathbb{R}$ be a risk-conditioned action value's quantile function. From Eq. (5), we define our optimal policy π^* as

$$\pi^* = \mathop{\text{argmax}}_\pi \mathop{\mathbb{E}}_{x\sim\phi}\left[\overleftarrow{Q}^\pi(x; s, \phi, \pi(s, \phi))\right] \quad (6)$$

for all $s \in \mathcal{S}$ and $\phi \in \Phi$.

As all $\phi \in \Phi$ constraints can be relaxed to the Monte-Carlo method (Choi et al. 2021; Yang, Sun, and Narasimhan 2019), the objective can be written as

$$\pi^* = \mathop{\text{argmax}}_\pi \mathop{\mathbb{E}}_{\phi\sim U[\Phi]}\left[\mathop{\mathbb{E}}_{x\sim\phi}\left[\overleftarrow{Q}^\pi(x; s, \phi, \pi(s, \phi))\right]\right] \quad (7)$$

Algorithm 1: GRIPS Training

$\pi_{\hat{\theta}}$: actor, $\overleftarrow{Q}_\theta$: critic, $\mathcal{B}$: Replay Buffer, env: Environment
enc, dec: encoder and decoder of risk proposal network
κ: cut-off parameter
while not converged **do**
  /* Sample $\mathbf{z}$ for roll out */
  $\mathbf{z}_{\text{r.o.}} \leftarrow (z_{\text{r.o.}}, c)$ where $z_{\text{r.o.}} \sim U[0, 1]^N, c \sim U[0, \kappa]$
  roll_out($\mathcal{B}$, env, $\pi_{\hat{\theta}}(\cdot, \mathbf{z}_{\text{r.o.}})$)
  /* Discard $\mathbf{z}_{\text{r.o.}}$ and sample $\mathbf{z}_{\text{tr.}}$ for RL update */
  $\mathbf{z}_{\text{tr.}} \leftarrow (z_{\text{tr.}}, c)$ where $z_{\text{tr.}} \sim U[0, 1]^N, c \sim U[0, \kappa]$
  $(s_t, a_t, r_t, s_{t+1}) \leftarrow$ sample($\mathcal{B}$)
  $x_i, x_j \sim U[0, 1]$, $a_{t+1} \leftarrow \pi_{\hat{\theta}}(s_{t+1}, \mathbf{z}_{\text{tr.}})$
  $\delta_{ij} \leftarrow r_t + \overleftarrow{Q}_\theta(x_j; s_{t+1}, \mathbf{z}_{\text{tr.}}, a_{t+1}) - \overleftarrow{Q}_\theta(x_i; s_t, \mathbf{z}_{\text{tr.}}, a_t)$
  critic_update(θ, $\nabla_\theta \mathcal{L}_{QR}(\delta_{ij})$) // $\mathcal{L}_{QR}$ in Eq. (1)
  $\overleftarrow{\phi}(x_i) \leftarrow dec(x_i; z_{\text{tr.}}, c)$ // inv. trans, see Eq. (4)
  $\mathcal{L}_\pi \leftarrow \overleftarrow{Q}_\theta(\overleftarrow{\phi}(x_i); s_t, \mathbf{z}_{\text{tr.}}, \pi_{\hat{\theta}}(s_t, \mathbf{z}_{\text{tr.}}))$ // in Eq. (6)
  actor_update($\hat{\theta}$, $-\nabla_{\hat{\theta}}\mathcal{L}_\pi$) // gradient ascent
end while

For uniform sampling on risk measures, denoted by $\phi \sim U[\Phi]$, we exploit a generative model. We also utilize the (1-)Wasserstein distance on the risk measures Φ, i.e.,

$$D_W(X, Y) = \int_0^1 |\overleftarrow{X}(x) - \overleftarrow{Y}(x)|dx \quad (8)$$

for $X, Y \in \Phi$. For the distance D_W in Eq. (8), the generative model is required to reproduce the quantile functions of risk measures $\overleftarrow{\phi}$; it is because D_W coincides with the L^1 distance on the space of quantile functions.

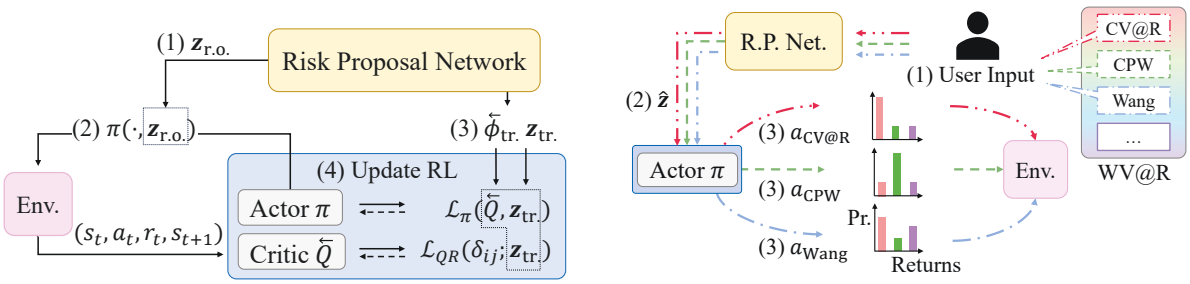
4 Approach

4.1 Overall Approach

To obtain a risk-conditioned RL agent through the objective in Eq. (6), our work employs a risk-conditioned actor-critic agent and a risk proposal network. The actor-critic agent is responsible for risk-conditioned decision-making, while the risk proposal network is responsible for generating and embedding risk measures, being a conditional generative model. Specifically, the validity of generated risk measures is controlled by a non-crossing quantile regressor, a component of the risk proposal network. We name our framework including the actor-critic agent and the risk proposal network, **GRIPS** (**G**eneralizing **RI**sk conditioned **P**olicy based on ri**S**k proposal network).

For risk-conditioned actions and values, the risk measure embedding $\mathbf{z}$ is considered an augmented state. Since excessively optimistic risk measures might cause numerical stability issues, we decompose $\mathbf{z}$ into two components (z, c) for conditional generation. Here, $z \sim U[0, 1]^N$ is a shape parameter and $c \sim U[0, \kappa]$ is a risk preference parameter. Note that $\kappa \in [0, 1]$ is referred to as a cut-off hyper-parameter, which controls c by

$$\mathbb{E}[\overleftarrow{\phi}] \approx c \leq \kappa \quad (9)$$



(a) Training Phase (b) Deployment Phase

Figure 1: Training and Deployment in GRIPS

where $\overleftarrow{\phi}$ is a decoded risk measure from $\mathbf{z} = (z, c)$. The details about the decomposition will be explained in Section 4.2. The actor takes the state tuple $(s, \mathbf{z})$ as input, generating action a, and the critic takes $(s, \mathbf{z}, a)$ and x_i as input, generating the quantile of a state-action value $\overleftarrow{Q}^\pi(s, \mathbf{z}, a; x_i)$ for risk measure embeddings $\mathbf{z}$ and quantile-level x_i.

During the training phase when an agent interacts with the environment and collects data (i.e., roll-out), risk measure embeddings $\mathbf{z}_{\text{r.o.}}$ are sampled for each episode and given to the agent[1]. When computing the RL loss, $\mathbf{z}_{\text{tr.}}$ is resampled, but $\mathbf{z}_{\text{r.o.}}$ is discarded, similar to RCDSAC; it discards roll-out risk measure embeddings and resamples the embeddings for training (Choi et al. 2021). Note that both zr.o and ztr. are sampled to prevent overly optimistic risk preferences upon decoding, with $c \sim U[0, \kappa]$. Finally, during the deployment phase when the agent is deployed, the risk proposal network encodes the user's input risk measure $\overleftarrow{\phi}$ in $\hat{\mathbf{z}}$. It is used to infer risk-conditioned actions, enabling the agent to behave in accordance with the risk measure specified by the user's input. Figure 1 depicts these agent training and deployment phases, and Algorithm 1 lists the training steps.

4.2 Risk Proposal Network

For effective agent training in the GRIPS framework, it is essential to have a model that can not only embed risk measures but also conditionally generate risk measures. The model capable of these functions is a variational-bayesian model (Kingma and Ba 2015). Specifically, we employ an AAE model (Makhzani et al. 2016), considering its flexibility in accommodating any prior distribution.

To overcome the issue, we use the methodology of WGAN-GP (Gulrajani et al. 2017). Furthermore, to mitigate the potential issue of sampling risk measures with excessively optimistic risk preferences, which can have adverse effects on agent training, we incorporate a conditional sampling mechanism into the AAE model. This allows for quantitative sampling based on specified risk preferences c in Eq. (9). This enhanced AAE model is named *Risk Proposal Network*. Algorithm 2 illustrates its training procedure.

Prior and Posterior: As discussed in Section 3.2, the posterior's support is quantile functions of risk measures. Specifically, the posterior space $\overleftarrow{\Phi}$ of the risk proposal network is defined as

$$\overleftarrow{\Phi} = \{\overleftarrow{\phi} : [0,1] \to [0,1] | x_1 < x_2 \Rightarrow \phi(x_1) \leq \phi(x_2)\}. \quad (10)$$

Note that $\overleftarrow{\Phi}$ is a compact space; thus, the prior space of the risk proposal network should be compact, as the encoder of risk-proposal network is a continuous map, which is implemented by a neural network. Accordingly, the prior distribution $U[0,1]^N$, which is compact and the simplest, is used.

To represent the risk measure $\overleftarrow{\phi}$ as input to the neural network (the risk proposal network), we use the sorted pairs $\{(x_i, y_i)\}_{i=1}^K$ such as

$$\{(x_i, y_i) \in [0,1]^2 | i < j \Rightarrow x_i \leq x_j \text{ and } y_i \leq y_j\}_{i=1}^K \approx \overleftarrow{\phi} \quad (11)$$

for $i, j = 1, \ldots, K$. To prevent the mode collapse, we sample $y_i = \overleftarrow{\phi}(x_i)$ in a hierarchical manner of $y_i \sim U[m, M]$, where $m \sim U[0,1]$ and $M \sim U[m, 1]$, while we have $x_i \sim U[0,1]$.

Architecture: The risk proposal network includes three modules: encoder, decoder, and discriminator. Let w be its model parameters. The encoder enc_w is responsible for representing risk measures in a form, which is similar to the prior distribution $U[0,1]^N$. It receives approximated $\overleftarrow{\phi}$ as input (in Eq. (11)) and returns the encoded representation $\hat{z}$ and risk preference $c \in [0,1]$ (in Section 4.1). The decoder dec_w is responsible for reconstructing or generating from the prior distribution. It receives z, c, and newly sampled points (i.e., quantile-level) $\{\hat{x}_i \sim U[0,1]\}_{i=1}^K$, yielding $\{\hat{y}_i \approx \overleftarrow{\phi}(x_i)\}_{i=1}^K$. The discriminator ensures that the encoder learns the prior distribution.

Loss Design: In order for the risk proposal network to be conditioned on risk preferences, we employ the conditional AAE loss defined as

$$\mathcal{L}_{AAE}(w) = \mathcal{L}_{\text{recon}}(w) + \mathcal{L}_{\text{discr}}(w) + \mathcal{L}_{\text{cond}}(w) \quad (12)$$

where $\mathcal{L}_{\text{recon}}$ is a reconstruction loss, $\mathcal{L}_{\text{discr}}$ is a prior distribution matching loss, and $\mathcal{L}_{\text{cond}}$ is a conditional sampling loss.

To ensure the reconstruction and the continuity between an embedding z and the input (and the output) $\overleftarrow{\phi}$, we use the quantile regression loss $\mathcal{L}_{\text{recon}}$ (in Eq. (1)) such as

$$\mathcal{L}_{\text{recon}}(w) = \mathcal{L}_{QR}(\delta_{ij}) \text{ where } \delta_{ij} = y_j - \hat{y}_i. \quad (13)$$

[1]Here r.o. and tr. denote roll-out and training, respectively.

Algorithm 2: Risk Proposal Network Training

enc_w: encoder, dec_w: decoder, $f_{\hat{w}}$: discriminator
η: gradient penalty coefficient, K: the num. of points
$m \sim U[0,1], M \sim U[m,1], \{y_i \sim U[m,M]\}$
$x_i \sim \text{sort}(U[0,1]), y_i \leftarrow \text{sort}(y_i), \hat{x}_j \sim U[0,1]$
/* Update AAE */
$\hat{z}, c \leftarrow enc_w(\{(x_i,y_i)\}_{i=1}^K)$ // $\overleftarrow{\phi} \approx \{(x_i,y_i)\}_{i=1}^K$
$\hat{y}_j \leftarrow dec_w(\hat{x}_j; \hat{z}, c)$ for $j = 1 \dots K$
$\delta_{ij} \leftarrow y_i - \hat{y}_j$, $\mathcal{L}_{\text{recon}} \leftarrow \mathcal{L}_{QR}(\delta_{ij})$ //recon loss in Eq. (13)
$\mathcal{L}_{\text{discr}} \leftarrow -f_{\hat{w}}(z)$, $\mathcal{L}_{\text{cond}} \leftarrow (c - \mathbb{E}[\hat{y}_j])^2$ // Eq. (14), (16)
$\mathcal{L}_{\text{AAE}} \leftarrow \mathcal{L}_{recon} + \mathcal{L}_{\text{discr}} + \mathcal{L}_{\text{cond}}$
$w \leftarrow \text{update}(w, \nabla_w \mathcal{L})$
/* Update discriminator */
$z \sim U[0,1]^N$, $D_W \leftarrow f_{\hat{w}}(z) - f_{\hat{w}}(\hat{z})$ // Eq. (15)
$z' \leftarrow \rho z + (1-\rho)\hat{z}$ where $\rho \sim U[0,1]$ // interpolate
$\mathcal{L}_{\text{WGAN}} \leftarrow -D_W + \eta \|\nabla_{z'} f_{\hat{w}}(z') - 1\|^2$ // Eq. (17)
$\hat{w} \leftarrow \text{update}(\hat{w}, \nabla_{\hat{w}} \mathcal{L}_{\text{WGAN}})$

As the embedding space $\mathcal{Z} := \{\hat{z} \in [0,1]^N | \hat{z} = enc_w(\overleftarrow{\phi})\}$ should have same probability measure as the prior distribution, we also use the prior distribution matching loss $\mathcal{L}_{\text{discr}}$ defined as

$$\mathcal{L}_{\text{discr}}(w) = -\mathbb{E}[f_{\hat{w}}(\hat{z} = enc_w(\overleftarrow{\phi}))] \tag{14}$$

where

$$f_{\hat{w}} = \underset{\|\nabla_z f\| \le 1}{\text{argsup}} \underbrace{\mathbb{E}_{z \sim U[0,1]^N}[f(z)]}_{\text{maximize } z \text{ from prior}} - \underbrace{\mathbb{E}_{\hat{z} \sim enc_w(\overleftarrow{\phi})}[f(\hat{z})]}_{\text{minimize } \hat{z} \text{ from encoding}}. \tag{15}$$

Here, $f_{\hat{w}}$ is a learned discriminator network parameterized by $\hat{w}$. Eq. (15) is a dual formulation of the Wasserstein distance (Villani et al. 2009) used when Eq. (8) is not feasible.

For the conditional sampling, we use $\mathcal{L}_{\text{cond}}$ defined as

$$\mathcal{L}_{\text{cond}}(w) = (c - \mathbb{E}[\hat{y}_j = dec_w(\hat{x}_i; \hat{z}, c)])^2. \tag{16}$$

This loss ensures that the decoder produces a risk measure ϕ conforming to a specified risk preference; $\mathbb{E}[\overleftarrow{\phi}]$, given z, c. Note that $\mathcal{L}_{\text{cond}}$ in Eq. (16) achieves this by utilizing an L^2 loss between $\mathbb{E}[\phi] = \mathbb{E}[\hat{y}_j]$ and c. The L^2 loss essentially enforces expectation matching.

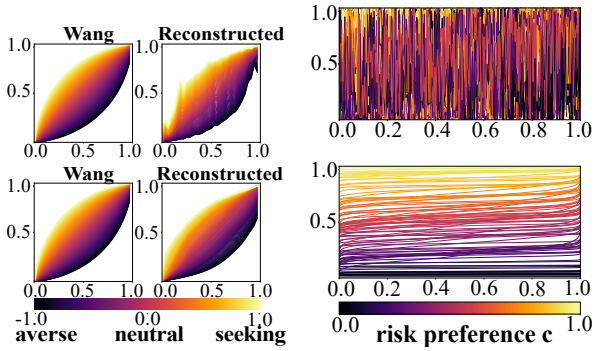
To satisfy the dual formulation of the Wasserstein distance in Eq. (15), the discriminator $f_{\hat{w}}$ is trained through

$$\mathcal{L}_{\text{WGAN}}(\hat{w}) = f_{\hat{w}}(\hat{z}) - f_{\hat{w}}(z) + \eta \|\nabla_{z'} f_{\hat{w}}(z') - 1\|^2 \tag{17}$$

where $z' = \rho z + (1-\rho)\hat{z}$ and $\rho \sim U[0,1]$. Here, z' is an interpolation between real and generated sample. The term $f_{\hat{w}}(\hat{z}) - f_{\hat{w}}(z)$ implements the gradient ascent in Eq. (15), while the other term $\|\nabla_{z'} f_{\hat{w}}(z') - 1\|^2$ implements the gradient penalty. When the discriminator $f_{\hat{w}}$ violates the 1-Lipschitz constraint $\|\nabla_z f\| \le 1$, the penalty is given. Empirically, this penalty for $\|\nabla_z f_{\hat{w}}\| < 1$ achieves robustness (Gulrajani et al. 2017). The gradient penalty coefficient $\eta > 0$ is a hyper-parameter.

4.3 Non-Crossing Quantile Regression

To ensure the effective learning of a risk-conditioned RL agent from generated risk measures, we address the crossing quantile problem, which requires the monotone constraint presented in Eq. (10). We use a conditional bijective normalizing flow (Rezende and Mohamed 2015; Winkler et al. 2019). We also demonstrate that the piecewise-linear normalizing flow is sufficient to approximate arbitrary risk measures within any error bound.



(a) Reconstruction (b) Random Samples

Figure 2: Reconstructed and Random Sampled Risk Measures without (top) and with (bottom) Non-Crossing Quantile Regressor. See Eq. (21) for the details on (a).

Without the non-crossing quantile regression, reconstructed risk measures contain numerous violating points (in Figure 2(a)) and randomly generated risk measures become useless (in Figure 2(b)). We empirically validate its importance in Section 5.4.

We use this normalizing flow model to implement a non-crossing quantile regressor. The reason why we choose normalizing flow is it is a bijective model. Therefore, it supports the monotone property; the continuous-bijection from $\mathbb{R}$ to $\mathbb{R}$ implies a strict monotone property (Wade 2010). We also leverage the model for the critics of risk-conditioned RL.

Architecture: The non-crossing quantile regressor includes a Gaussian quantile function, a piecewise-linear transform, and a sigmoid function, as shown in Figure 3. The Gaussian quantile function processes $\hat{x}_i \mapsto F^{-1}_{\mathcal{N}(0,1)}(\hat{x}_i) =: x_i$ (Def. 1), while it is agnostic to an embedding of risk-measure $\mathbf{z}$. Then, the piecewise-linear transform processes x_i[2]. Specifically, the piecewise-linear transform includes a sequence of affine and PReLU layers (He et al. 2015; Dinh, Sohl-Dickstein, and Bengio 2016). Each affine layer receives $\mathbf{z}$ and infers the scale $\varsigma > 0$ and bias $b \in \mathbb{R}$. The evaluated transform is $x_i \mapsto (\varsigma x_i + b)$. A PReLU layer receives $\mathbf{z}$ and infers scale $\hat{\varsigma} > 0$. The transform is $x_i \mapsto (\mathbb{I}(x > 0)x_i + \hat{\varsigma}\mathbb{I}(x_i \le 0)x_i)$. These two transforms are sequentially applied to x_i. Finally, the sigmoid function aligns the output $\hat{y}_i$ with the codomain, $[0,1]$, as described in Eq. (10).

Approximation Property: Now we discuss that piecewise-linear transform can approximate any risk measures within any error $\varepsilon > 0$ with respect to the Wasserstein distance.

Proposition 1. *Let $\phi : [0,1] \to [0,\infty]$ be any risk measure.*

[2]We override the symbol x_i to represent an intermediate value between $\hat{x}_i$ and $\hat{y}_i$ following Notation 1, as x_i is transformed from $\hat{x}_i$ via the previous layer in a bijective way.

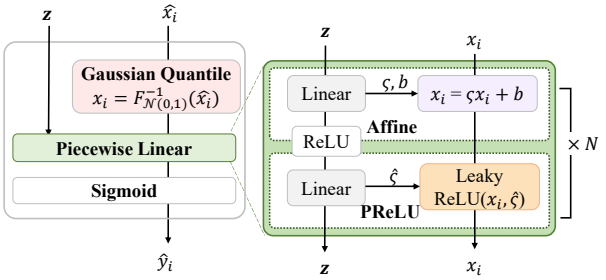


Figure 3: Non-Crossing Quantile Regressor. Each block represents a layer.

Given $\varepsilon > 0$, there exists a piecewise-linear approximation of $\overleftarrow{\phi}$ denoted as $\overleftarrow{\psi}$ such that $\int_0^1 |\overleftarrow{\phi} - \overleftarrow{\psi}| d\lambda < \varepsilon$ where λ is the Lebesgue measure on $[0, 1]$.

We provide a proof for the cases when risk measure ϕ is a conventional density function (i.e., ϕ has no atom). The other cases are provided in Appendix.

Proof: We show that the density's total variation is bounded by ε, since the total variation is an upper bound of the Wasserstein distance on a connected interval (Villani et al. 2009). ϕ is Lebesgue integrable in this case. By the definition of Lebesgue integral, there is a sequence of simple functions[3] $\{\psi_n\}_{n\in\mathbb{N}}$ such that $\psi_n \le \psi_{n+1} \le \phi, \forall n \in \mathbb{N}$, and $\lim_{n\to\infty} \int_0^1 \psi_n d\lambda = \int_0^1 \phi d\lambda$. Given $\varepsilon > 0$, we choose $N \in \mathbb{N}$ such that $n \ge N$ implies $\int_0^1 |\phi - \psi_n| d\lambda < \frac{1}{2}\varepsilon$.

Let $\psi := \left(1 - \int_0^1 \psi_N d\lambda\right) + \psi_N$. Then, we obtain

$$\begin{aligned} \int_0^1 |\psi - \phi| d\lambda &= \int_0^1 |\psi - \psi_N + \psi_N - \phi| d\lambda \\ &\le \int_0^1 |\psi - \psi_N| d\lambda + \int_0^1 |\psi_N - \phi| d\lambda < \frac{1}{2}\varepsilon + \frac{1}{2}\varepsilon = \varepsilon. \end{aligned} \tag{18}$$

Since ψ is simple density function, its quantile $\overleftarrow{\psi}$ is piecewise-linear.

5 Experiment

In this section, we evaluate our GRIPS for arbitrary *WV@R* risk measures, aiming to clarify whether it can adapt to given risk measures without explicit information about them. Surprisingly, GRIPS often surpasses IQN, which is trained to fit to a specific risk measure. we hypothesize that the abundant exploration leads to GRIPS's robust performance because GRIPS experiences more diverse episodes during training from randomized risk measures, compared to IQN. To verify the hypothesis, we perform experiments in slightly modified Mujoco environments, which are well-known environments. We also have ablation studies highlighting the impact of the non-crossing quantile regressor.

[3] a function whose codomain is a finite set.

5.1 Experimental Setup

Evaluation: We accumulate scores from 100 (self-driving) to 1000 (others) episodes for each seed, computing their weighted average according to evaluation risk measures by Monte-Carlo method. The followings are the quantiles of risk measures we assess:

- *CV@R* denoted by $CV@R_{100\alpha\%}$ has the quantile:

$$\overleftarrow{\phi}_{\text{CV@R}}(x; \alpha) = \begin{cases} \alpha x & \text{if } \alpha \in [0, 1] \\ (2-\alpha)x + (\alpha - 1) & \text{if } \alpha \in (1, 2] \end{cases}. \tag{19}$$

  It is risk-averse for $\alpha \in [0, 1]$. For $\alpha \in (1, 2]$, it is risk-seeking (inverted *CV@R*) taking an average for upper $(\alpha - 1)$-quantile values.

- *CPW* denoted by CPW_α has the following quantile:

$$\overleftarrow{\phi}_{CPW}(x; \alpha) = \frac{x^\alpha}{(x^\alpha + (1-x)^\alpha)^{\frac{1}{\alpha}}}, \alpha \in (0, 1). \tag{20}$$

- *Wang* denoted by $Wang_\alpha$ has the following quantile:

$$\overleftarrow{\phi}_{Wang}(x; \alpha) = F_{\mathcal{N}(0,1)}(F^{-1}_{\mathcal{N}(0,1)}(x) + \alpha), \alpha \in \mathbb{R} \tag{21}$$

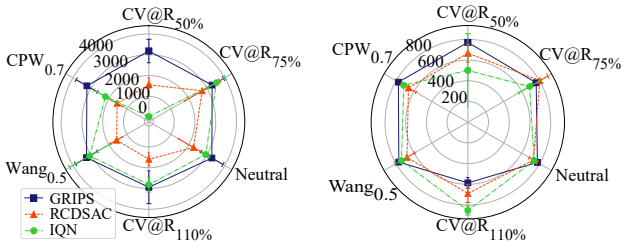
  where $F_{\mathcal{N}(0,1)}$ and $F^{-1}_{\mathcal{N}(0,1)}$ are the cumulative density and quantile of the Gaussian distribution (Def. 1).

Baselines: We compare GRIPS with several baselines.

- IQN is trained with explicitly provided risk measures and evaluated using the same measures.
- RCDSAC is trained for a target risk set without specific confidence or parameter details. The risk measures should be a closed form of a function defined by the parameter.
- GRIPS (ours) is trained for all *WV@R* risk measures except for excessively optimistic ones.

All algorithms are based on SAC (Haarnoja et al. 2018), especially TQC (Kuznetsov et al. 2020) variants for actor-critic implementations.

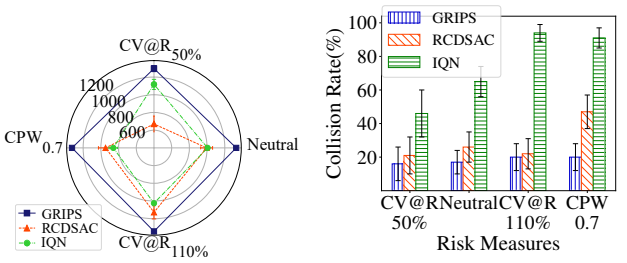
5.2 Financial Task



(a) Stock-Unhedge (b) Stock-Hedge

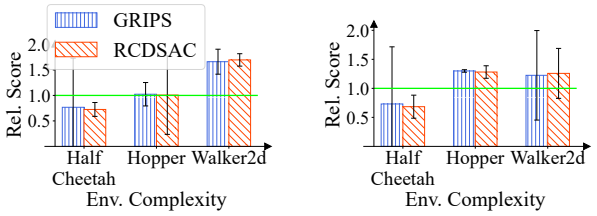
Figure 4: Finance Task Result

For the experiments in financial tasks, we utilize the MetaTrader simulator (Amin 2021). Specific information about stock price is given as a state, and asset price increments are given as a reward. An action is the ratio of buying or selling stocks. Stock-Unhedge is an environment with large volatility,



(a) Scores (b) Collision Rate

Figure 5: Self-Driving Task Result

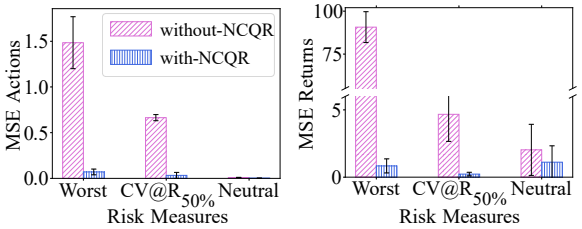


(a) $CV@R_{50\%}$ (b) $CV@R_{110\%}$

Figure 6: Mujoco Result. Y-axis is relative scores w.r.t. IQN.

actions	*Worst*	$CV@R_{50\%}$	*Neutral*
-1	$\mathbf{-1}$	-1	-1
0	-2	**0**	2
1	-12.5	-0.5	**12.5**

Table 1: Toy Environment Returns for Actions



(a) MSE Actions (b) MSE Rewards (Expectile). Expectile distortions are according to the labeled risk-measures.

Figure 7: Toy Env. Result. NCQR is the non-crossing quantile regressor. 'Without-NCQR' is IQN (i.e., cosine-embedding)

while Stock-Hedge is with relatively less volatility. In Figure 4, GRIPS exhibits comparable performance to IQN and RCDSAC. For Stock-Hedge, GRIPS shows $3.9\% \sim 54.0\%$ higher over IQN. Meanwhile, for Stock-Unhedge, GRIPS demonstrates a robust performance range from a decrease of 6.81% at its lowest to an increase of 47%, with the exception of $CV@R_{50\%}$ where the IQN exhibits an outlier score.

5.3 Self-Driving Task

For self-driving tasks, we use the Donkey Car simulator with AVC-Sparkfun map (Kramer 2023), where a state is the front view image, an action is throttling and steering, and a reward is the route adherence. In Figure 5(a), GRIPS exhibits a performance increase of $16\% \sim 54\%$ over IQN, verifying its adaptability to target risks. Figure 5(b) shows different behaviors even for the same target risk measures. IQN and RCDSAC exhibit large gaps in the collision rate, while their achieved scores remain closely comparable.

5.4 Empirical Analysis

Analysis of Generalizing Effect: We investigate whether generalizing risk measures can affect RL agents. We use Mujoco (HalfCheetah, Hopper, Walker2d) for which expert policies are well established. We inject noise to rewards to prevent a policy from converging to a deterministic solution. According to the expert policy's state distribution, the importance of exploration can be ordered; i.e., HalfCheetah $<$ Hopper $<$ Walker2D. Appendix includes the detailed justification. Compared to IQN, in Figure 6, both GRIPS and RCDSAC show a performance decrease of approximately 24% in the simpler environment, Halfcheetah. However, in the more complex environment, Walker2d, they outperform IQN with an improvement of about 53%. The results are consistent with the complex tasks where GRIPS outperforms IQN; i.e., Stock-Unhedge with $CV@R_{50\%}$ and the self-driving task. The extended exploration strategy, based on various risk measures during training, is attributed to the superiority of GRIPS. Conversely, when the exploration is insignificant, risk measures act as noises, resulting in performance degradation.

Ablation Study: To evaluate the effect of the non-crossing quantile regressor in the risk proposal network, we conduct experiments in a toy environment, where direct analytic solutions are given according to risk measures. Specifically, it is a single-step environment with a singleton state set and an one-dimensional action space, i.e., $\mathcal{A} = [-1, 1]$. We deliberately use such rewards whose optimal actions can be differently computed for specific risk measures. See Appendix and Table 1 for details. As observed in Figure 7, it is crucial to solve the crossing quantile problem. With the non-crossing quantile regressor (with-NCQR), MSE is consistently lower than the other (without-NCQR) for all risk measures. Conversely, for without-NCQR, only the *Neutral* case has lower errors, implying that generalized risk-conditioned policies are not established; only some specific risk measure is handled.

6 Conclusion and Limitation

In this paper, we presented a generalized, risk-conditioned RL framework to accommodate *WV@R*. We devised the risk proposal network using a conditional AAE to effectively generate risk measures and embed them for RL agent training. Using the normalizing flow, we also addressed the crossing quantile problem. Our framework focuses on *WV@R*, while not encompassing all risk measures, such as *CMV* (Vadori et al. 2020) or *variance*, but can cover a wide range of risk measures applicable to real-world problems.

Acknowledgements

This work was supported by the Institute for Information and Communications Technology Planning and Evaluation (IITP) under grant funded by the Korea government(MSIT) (No.2022-0-00043, No.2022-0-01045; Adaptive Personality for Intelligent Agents, Self-directed Multimodal Intelligence for solving unknown, open domain problems), by DNA+Drone Technology Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science and ICT(No. NRF-2020M3C1C2A01080819), by ICT Creative Consilience program supervised by the IITP under Grant IITP-2020-0-01821, and by the NRF grant funded by the MSIT (No. RS-2023-00213118).

References

Acerbi, C.; Nordio, C.; and Sirtori, C. 2001. Expected shortfall as a tool for financial risk management. *arXiv preprint cond-mat/0102304*.

Acerbi, C.; and Simonetti, P. 2002. Portfolio optimization with spectral measures of risk. *arXiv preprint cond-mat/0203607*.

Adam, A.; Houkari, M.; and Laurent, J.-P. 2008. Spectral risk measures and portfolio selection. *Journal of Banking & Finance*, 32(9): 1870–1882.

Amin, H. M. 2021. gym-mtsim. https://github.com/AminHP/gym-mtsim. Accessed: 2023-01-28, version 1.1.0.

BCBS. 2009. Revisions to the Basel II market risk framework. https://www.bis.org/publ/bcbs158.htm. Accessed: 2022-12-15.

Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, 449–458. PMLR.

Bondell, H. D.; Reich, B. J.; and Wang, H. 2010. Noncrossing quantile regression curve estimation. *Biometrika*, 97(4): 825–838.

Brando, A.; Center, B. S.; Rodriguez-Serrano, J.; Vitria, J.; et al. 2022. Deep Non-Crossing Quantiles through the Partial Derivative. In *International Conference on Artificial Intelligence and Statistics*, 7902–7914. PMLR.

Choi, J.; Dance, C.; Kim, J.-e.; Hwang, S.; and Park, K.-s. 2021. Risk-Conditioned Distributional Soft Actor-Critic for Risk-Sensitive Navigation. In *International Conference on Robotics and Automation (ICRA)*, 8337–8344. IEEE.

Chow, Y.; Ghavamzadeh, M.; Janson, L.; and Pavone, M. 2017. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1): 6070–6120.

Dabney, W.; Ostrovski, G.; Silver, D.; and Munos, R. 2018. Implicit quantile networks for distributional reinforcement learning. In *International Confernce on Machine Learning*, 1096–1105. PMLR.

Dette, H.; and Volgushev, S. 2008. Non-crossing nonparametric estimates of quantile curves. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70(3): 609–627.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, volume 30.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 1861–1870. PMLR.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

He, X. D.; Jin, H.; and Zhou, X. Y. 2015. Dynamic portfolio choice when risk is measured by weighted VaR. *Mathematics of Operations Research*, 40(3): 773–796.

Howard, R. A.; and Matheson, J. E. 1972. Risk-sensitive Markov decision processes. *Management science*, 18(7): 356–369.

Jabr, R. A. 2005. Robust self-scheduling under price uncertainty using conditional value-at-risk. *IEEE Transactions on Power Systems*, 20(4): 1852–1858.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *International Conference on Learning Representations*.

Kramer, T. 2023. donkey-car.

Kuznetsov, A.; Shvechikov, P.; Grishin, A.; and Vetrov, D. 2020. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning*, 5556–5566. PMLR.

Makhzani, A.; Shlens, J.; Jaitly, N.; and Goodfellow, I. 2016. Adversarial Autoencoders. In *International Conference on Learning Representations*. PMLR.

Mihatsch, O.; and Neuneier, R. 2002. Risk-sensitive reinforcement learning. *Machine learning*, 49: 267–290.

Pruzzo, L.; Cantet, R. J.; and Fioretti, C. C. 2003. Risk-adjusted expected return for selection decisions. *Journal of animal science*, 81(12): 2984–2988.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rahimi, M.; and Ghezavati, V. 2018. Sustainable multi-period reverse logistics network design and planning under uncertainty utilizing conditional value at risk (CVaR) for recycling construction and demolition waste. *Journal of cleaner production*, 172: 1567–1581.

Rezende, D.; and Mohamed, S. 2015. Variational inference with normalizing flows. In *International conference on machine learning*, 1530–1538. PMLR.

Sato, M.; Kimura, H.; and Kobayashi, S. 2001. TD algorithm for the variance of return and mean-variance reinforcement learning. *Transactions of the Japanese Society for Artificial Intelligence*, 16(3): 353–362.

Tamar, A.; Glassner, Y.; and Mannor, S. 2015. Optimizing the CVaR via sampling. In *Association for the Advancement of Artificial Intelligence*.

Vadori, N.; Ganesh, S.; Reddy, P.; and Veloso, M. 2020. Risk-sensitive reinforcement learning: A martingale approach to reward uncertainty. In *International Conference on Artificial Intelligence in Finance*, 1–9.

Villani, C.; et al. 2009. *Optimal transport: old and new*, volume 338. Springer.

Wade, W. 2010. *An Introduction to Analysis*. Pearson Education International. ISBN 9780136153702.

Wehenkel, A.; and Louppe, G. 2019. Unconstrained monotonic neural networks. In *Advances in Neural Information Processing Systems*, volume 32.

Winkler, C.; Worrall, D.; Hoogeboom, E.; and Welling, M. 2019. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*.

Yang, R.; Sun, X.; and Narasimhan, K. 2019. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *Advances in Neural Information Processing Systems*, volume 32.

Zhang, Y.; Wang, J.; Ding, T.; and Wang, X. 2018. Conditional value at risk-based stochastic unit commitment considering the uncertainty of wind power generation. *IET Generation, Transmission & Distribution*, 12(2): 482–489.

Zhou, F.; Wang, J.; and Feng, X. 2020. Non-crossing quantile regression for distributional reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 15909–15919.